**Problema**

O problema consiste em utilizar 27 atributos que descrevem o estado de saúde de cavalos e conseguir descobrir se o cavalo vai sobreviver ou morrer. Esse problema de classificação é apresentado em 2 arquivos, 1 para o treinamento dos modelos criados e outro para o teste dos modelos.

**Proposta**

Para o problema, foi proposto utilizar a linguagem de programação python devido a sua facilidade de implementação e por ser amplamente utilizada na criação de modelos peditórios. A IDE escolhida foi *a Jupyter Notebook* devido a sua praticidade para criar e debugar o código.

**Bibliotecas Utilizadas**

A principal biblioteca utilizada para a solução do problema é a *pandas*, porem as demais utilizadas estão abaixo.

**from** sklearn**.**model\_selection **import** train\_test\_split

**from** sklearn**.**neighbors **import** KNeighborsClassifier

**from** sklearn**.**feature\_selection **import** RFE

**from** sklearn**.**linear\_model **import** LogisticRegression

**from** sklearn **import** tree

**import** sklearn**.**metrics **as** metrics

**import** sklearn**.**preprocessing **as** preprocessing

**import** pandas **as** pd

**import** matplotlib**.**pyplot **as** plt

**import** numpy **as** np

pd**.**options**.**display**.**max\_rows **=** 999

**Arquivos Utilizados**

Os 2 arquivos utilizados (horse.csv e horseTest.csv) foram lidos diretamente via a função *read\_csv* da biblioteca *pandas*. Um é utilizado para a criação e treinamento dos modelos e o outro para o teste dos modelos.

#Load in the data with `read\_csv()`

horsesDataSet **=** pd**.**read\_csv**(**'horse.csv'**,** header**=**0**,** delimiter**=**','**)**

horsesDataSetTest **=** pd**.**read\_csv**(**"horseTest.csv"**,** header**=**0**,** delimiter**=**','**)**

#description of dataSet

descriptionHorsesDataSet **=** horsesDataSet**.**describe**(**include**=**'all'**)**

descriptionHorsesDataSetTest **=** horsesDataSetTest**.**describe**(**include**=**'all'**)**

Análise exploratória

Pré processamento

Missing Values

Atributos Desnecessários

Balanceamento

Transformação dos Atributos

Testes de modelos

KNN

KNN + LR

Decision tree

Decision tree + LR